

ФАЙЛОВЕ (ЧАСТ 2). ДВОИЧНИ ФАЙЛОВЕ.

Задача първа:

След като Явор успешно завърши първи курс на ФМИ, той беше готов да започне работа. Но понеже работодателите, търсещи неговите невероятни умения, бяха повече от очакваното, той реши да изтрие имейла си и да направи собствена система, чрез която да получава покани за работа.

Създайте структура, описваща обява за работа. Структурата трябва да съдържа:

1. Име на фирмата. То трябва да е с максимална дължина от 25 символа.
2. Брой на програмистите, с които той ще работи в екип.
3. Брой на дните платен отпуск през годината.
4. Сума, която ще му бъде изплатена след успешно завършване на проекта (от тип long long)

Създайте функция, която прочита n на брой заявки от стандартния вход. Заявките се запазват в **двоичен** файл, като ако във файла вече има информация, тя не се изтрива.

Създайте функция void filterOffers(const char* filePath, long long minSalary); която приема адрес на файл, от който четем заявките и минимална заплата. Извежда на стандартния изход всички обяви за работа, които предлагат поне толкова пари, колкото са подадени като аргумент.

Създайте функция, bool existOffer(const char* filePath, const char* name); която проверява дали дадена фирма е пратила покана за работа.

Бонус:

Създайте void perfectOffer(const char* filePath, int maxCoworkers, int minVacancyDays, int minSalary); която извежда във **файл** всички обяви, които отговарят точно на изискванията на Явор.

Бонус:**

Създайте система, която работи чрез следните заявки:

1. a <оферта> - добавя оферта за работа
2. i - показва всички оферти
3. s <име на фирма> - връща дали фирмата съществува
4. f <число> - извежда всички обяви, за които заплата е по - висока от числото
5. q - изход от системата